เขมะทัต วิภาตะวนิช

มารู้จักมาตรฐานระบบเครือข่ายคอมพิวเตอร์

กล่าวถึงคำจำกัดความ สถาปัตยกรรมลำดับชั้น ตามข้อกำหนด ISO และรูปแบบโปรโตคอล HDLC

ระบบเครือข่าย เมื่อพิจารณาตามอัตราความเร็วในการส่งผ่านข้อมูล ระยะทาง และลักษณะในการให้บริการจะแบ่งเป็น 2 ประเภทใหญ่ ๆ คือ เครือข่ายเฉพาะบริเวณ (Local Area Network (LAN)) และเครือข่ายบริเวณกว้าง (Wide Area Network (WAN)) โดยเครือข่ายเฉพาะบริเวณจะมีอัตราเร็วในการส่งผ่านข้อมูลค่อนข้างสูง คือประมาณ 4-100 เมกะบิต/วินาที ระยะทางอยู่ในช่วงประมาณ 2.5-200 กิโลเมตร สำหรับการให้บริการนั้นผู้ที่เป็นเจ้าของระบบจะเป็นผู้กำหนด ซึ่งอาจจะเป็นเอกชนหรือรัฐบาลก็ได้ มาตรฐานและเครือข่ายเฉพาะบริเวณ เช่น IEEE 802.3, 802.4, 802.5 และ FDDI (Fiber Distributed Data Interface) เป็นต้น เครือข่ายบริเวณกว้างจะมีอัตราเร็วในการส่งผ่านข้อมูลเมื่อถึงผู้ใช้บริการไม่สูงมาก คือ ประมาณ 1,200 บิต/วินาที-2 เมกะบิต/วินาที (2.048 เมกะบิต/วินาที) ระยะที่ใกล้โดยเฉลี่ย 8 กิโลเมตร ส่วนระยะไกลก็อาจจะถึงหลายหมื่นกิโลเมตร สำหรับการให้บริการนั้นจะครอบคลุมภายในประเทศและติดต่อกับต่างประเทศ จึงจำเป็นต้องมีองค์กรของรัฐเข้ามาเกี่ยวข้องด้วย มาตรฐานของเครือข่ายบริเวณกว้าง เช่น X.25, X.75 เป็นต้น

สำหรับบทความนี้จะกล่าวถึงมาตรฐานที่เกี่ยวข้องเครือข่าย โดยจะ

เน้นที่คำจำกัดความ สถาปัตยกรรมลำดับชั้นตามข้อกำหนดของ ISO และในช่วงท้ายจะเป็นรูปแบบโปรโตคอล HDLC ซึ่งเป็นโปรโตคอลชั้นที่ 2 ที่ถูกกำหนดโดย ISO สำหรับบทแทรกนั้นจะเป็นสิ่งที่โปรโตคอลในระบบสื่อสารข้อมูลจะต้องมี ซึ่งก็คือ Sliding window และการคำนวณ FCS

มาตรฐานมีไว้เพื่ออะไร

ถ้าพิจารณาดูจะพบว่า ในตลาดของเครื่องคอมพิวเตอร์นั้นมีอยู่มากมายหลายเครื่องหมายการค้า บางเครื่องหมายการค้าอาจจะคล้ายกัน แต่บางเครื่องหมายการค้าก็แตกต่างกันอย่างสิ้นเชิง เครื่องคอมพิวเตอร์ที่มีลักษณะคล้ายกันนั้นมักจะอยู่ในระดับที่เรียกว่าไมโครคอมพิวเตอร์สำหรับเครื่องในระดับมินิ เมนเฟรม หรือซูเปอร์คอมพิวเตอร์แล้ว ถึงแม้ว่าจะมีก็เป็นจำนวนไม่มากนัก ในการเชื่อมโยงเครื่องคอมพิวเตอร์เข้าหากันเพื่อทำเป็นเครือข่าย จะเป็นการยากอย่างยิ่งถ้าเรานำเครื่องคอมพิวเตอร์ต่างเครื่องหมายการค้ากันมาเชื่อมโยงเข้าหากัน ทั้งนี้เพราะว่า บริษัทผู้ผลิตแต่ละรายย่อมมีวิธีการในการเชื่อมต่อที่ไม่เหมือนกัน วิธีการที่ว่านี้ไม่เพียงแต่เฉพาะฮาร์ดแวร์เท่านั้น แต่ยังรวมถึงซอฟต์แวร์ด้วย เมื่อเรามองเข้าสู่สเกล (scale) ที่ใหญ่ขึ้นนั่นก็คือ การเชื่อมต่อเครือข่ายระหว่างประเทศทั่วโลก ปัญหาที่เกิดขึ้นก็ยังคงเหมือนเดิมและถ้าเป็นเช่นนี้ตลอดไป ปัญหาที่เกิดขึ้นย่อมจะตกอยู่กับผู้ใช้งาน ทำให้ผู้ใช้งานไม่สามารถเลือกใช้เครื่องคอมพิวเตอร์ได้อย่างคล่องตัว และในอนาคตก็ไม่มีหลักประกันที่แน่นอนว่าผลิตภัณฑ์ของบริษัทนั้น ๆ จะยังคงมีตลอดไป ดังนั้นจึงมีหลายองค์กรไม่ว่าจะเป็นภายในหรือระหว่างประเทศ เช่น IEEE (The Institute of Electrical and Electronics Engineers), ISO (International Standard for Organization), CCITT* เป็นต้น พยายามที่จะเป็นตัวกลางเสมือนเป็นผู้ควบคุมดูแลผลประโยชน์ของทั้งผู้ผลิตและผู้ใช้งาน จึงได้กำหนดมาตรฐานที่ใช้ในการเชื่อมโยงเครือข่ายหรือคอมพิวเตอร์เข้าหากันซึ่งเป็นที่ยอมรับกันทั่วโลก ทำให้ผู้ใช้มีความคล่องตัวในการเลือกซื้อผลิตภัณฑ์มาใช้งานให้ได้ประโยชน์สูงสุด เช่น การเชื่อมต่อเครื่องคอมพิวเตอร์เข้าหากันโดยผ่านชุมสายโทรศัพท์ก็ต้องใช้อุปกรณ์ที่เรียกว่า โมเด็ม (modem) เพื่อใช้ในการแปลงสัญญาณดิจิตอลเป็นสัญญาณอะนาล็อก หรือโดยนัยกลับกัน มาตรฐานของโมเด็มที่ใช้กันอยู่นี้ ก็เป็นไปตามมาตรฐานของ Bell หรือ CCITT V-series เช่น V.22 bis, V.32 เป็นต้น นอกจากนี้การเชื่อมต่อระหว่างโมเด็มกับคอมพิวเตอร์ก็ใช้มาตรฐานที่เรียกว่า CCITT V.24 หรือที่รู้จักกันโดยทั่วไปว่า RS-232C (ถึงแม้ว่าจะมีข้อกำหนดที่แตกต่างกันออกไปบ้างก็ตาม) ซึ่งเป็นมาตรฐานของ EIA (Electronic Industries Association) ในที่นี้ขอให้ลองพิจารณาระบบโทรศัพท์ที่ใช้กันอยู่ทุกวันนี้ ในอดีตจะเห็นได้ว่า ถ้าต้องการติดต่อไปยังปลายทางจะใช้วิธีหมุนหมายเลขซึ่งก็คือ ระบบพัลส์ (pluse) นั่นเอง ในปัจจุบันเราเพียงแต่กดหมายเลขที่ต้องการก็สามารถติดต่อไปยังปลายทางได้เช่นเดียวกัน ซึ่งก็คือระบบ DTMF (Dual Tone Multi-Frequency) ถ้านำระบบทั้งสองมาสลับกันใช้จะเห็นได้ว่าไม่สามารถทำงานได้ ซึ่งเปรียบเทียบได้กับการใช้มาตรฐานที่แตกต่างกันนั่นเอง ดังนั้นจะเห็นได้ว่าระบบที่ให้บริการในลักษณะที่เป็นสาธารณะนั้น มาตรฐานย่อมเป็นสิ่งที่ไม่อาจละเลยได้

สถาปัตยกรรมลำดับชั้น

ถ้าพิจารณาเหตุการณ์ที่เกิดขึ้นในการส่งผ่านแฟ้มข้อมูลระหว่างเครื่องคอมพิวเตอร์ต่างเครื่องหมายการค้ากันโดยผ่านเครือข่ายคอมพิวเตอร์ จะพบว่ามีปัญหาเกิดขึ้นหลายประการด้วยกัน เช่น โปรเซสที่อยู่บนเครื่องคอมพิวเตอร์ต้นทาง จะหาตำแหน่งและเส้นทางของโปรเซสที่ต้องการจะติดต่อด้วยที่ปลายทางได้อย่างไร ถ้าระบบเกิดความผิดพลาดขึ้นขณะที่มีการส่งผ่านข้อมูล ซึ่งอาจเนื่องมาจากเครื่องคอมพิวเตอร์ทำงานผิดพลาดจนต้องหยุดการทำงานของตัวเอง จะมีวิธีการอย่างไรเพื่อที่จะทำให้ระบบกลับคืนสู่สภาพเดิม และพร้อมกันนั้นจะแน่ใจได้อย่างไรว่าข้อมูลที่ส่งผ่านกันมีการเรียงลำดับของข้อมูลตลอดจนเรียงลำดับบิดก่อนหลังได้ถูกต้อง จะเห็นได้ว่าปัญหาดังกล่าวมีขนาดใหญ่มาก และจะเป็นการยากยิ่งถ้าจะพยายามสร้างวัตถุ (object) ใด ๆ ขึ้นมาวัตถุหนึ่ง เพื่อแก้ปัญหาทั้งหมดที่เกิดขึ้น ดังนั้นจึงได้เกิดแนวความคิดที่จะแบ่งหน้าที่การแก้ปัญหาออกเป็นส่วนย่อยหลาย ๆ ส่วน พร้อมกันนั้นก็จัดระเบียบของปัญหาที่เกิดขึ้นให้อยู่ในกลุ่มเดียวกัน จากหลักการนี้เองจึงได้เกิดสถาปัตยกรรมลำดับชั้นขึ้น โดยแบ่งระดับชั้นของการเชื่อมต่อระหว่างคอมพิวเตอร์ออกเป็น n ชั้น แต่ละชั้นจะมีหน้าที่เฉพาะอย่าง ดังแสดงในรูปที่ 1 การเชื่อมต่อจะเกิดขึ้นใน 2 ลักษณะ คือ แนวนอนและแนวตั้ง แนวนอนจะเป็นการสื่อสารระหว่างต้นทางกับปลายทางที่อยู่ใน

* CCITT = Comité Conmltatif Internationale de Télégraphique et Teléphonique

ระดับเดียวกัน โดยใช้โปรโตคอลในระดับนั้น ๆ เป็นตัวจัดการ ส่วนแนวตั้งจะเป็นการขอและให้บริการผ่านจุดเชื่อมต่อระหว่างชั้นที่อยู่ติดกันเท่านั้น

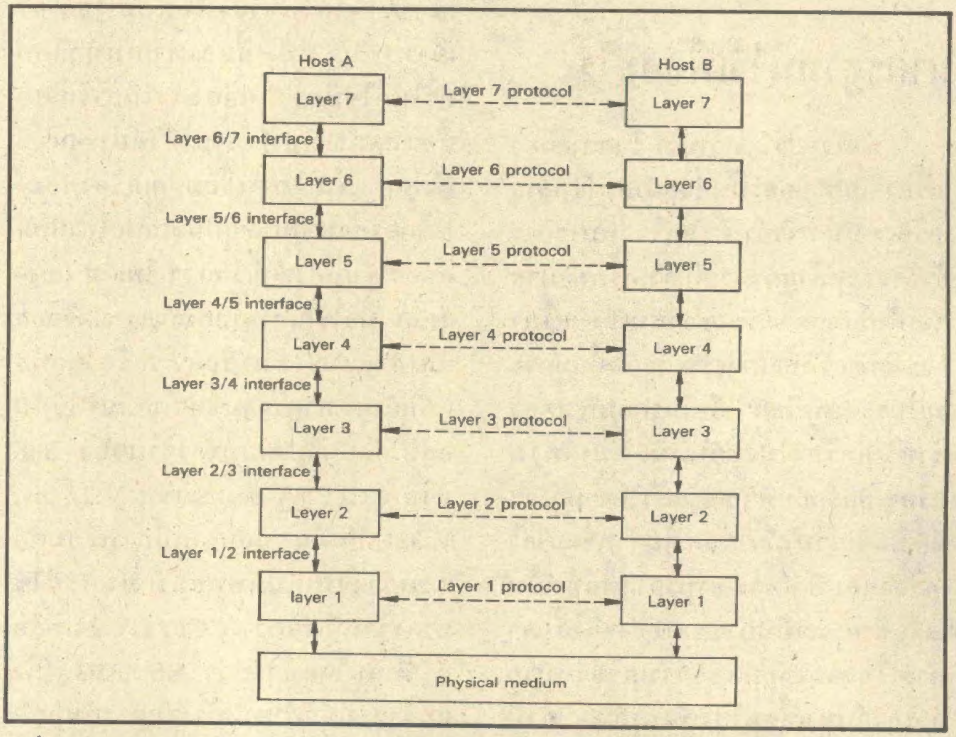
รูปแบบอ้างอิงมาตรฐาน (Reference Model)

จากการที่มีการนำคอมพิวเตอร์เข้ามาช่วยงานในหลายลักษณะ และมีแนวโน้มที่จะขยายการใช้งานให้กว้างขวางมากขึ้นในอนาคต ทำให้ผู้ใช้มีความต้องการที่จะเชื่อมต่อเครื่องคอมพิวเตอร์เพื่อสื่อสารถึงกันมากขึ้นเป็นเงาตามตัว คณะกรรมการของ ISO ชุดที่ 97 (Technical Committee 97 (TC 97)) ที่ดูแลทางด้านการประมวลผลระบบสารสนเทศได้ตระหนักถึงความต้องการเช่นนี้เป็นอย่างดี จึงได้ตั้งคณะทำงานที่เรียกว่า คณะกรรมการย่อยชุดที่ 16 (Subcomittee 16 (SC16)) เพื่อศึกษาและกำหนดรูปแบบอ้างอิงมาตรฐานที่ใช้ในการเชื่อมโยงเครื่องคอมพิวเตอร์หลาย ๆ ชนิดเข้าหากัน (heterogeneous network) และในช่วงฤดูใบไม้ผลิปี ค.ศ. 1983 ได้มีการประกาศรูปแบบอ้างอิงมาตรฐานของ ISO โดยใช้รหัสว่า ISO 7498 ซึ่งเป็นรูปแบบที่กล่าวถึงการเชื่อมต่อเครื่องคอมพิวเตอร์เข้าหาด้วยกันในลักษณะที่เรียกว่าระบบเปิด (Open System Interconnection (OSI)) ซึ่งประกอบด้วย 2 ส่วนใหญ่ ๆ คือ องค์ประกอบของสถาปัตยกรรมลำดับชั้น ซึ่งกล่าวถึงส่วนประกอบที่ถูกกำหนดหรือสร้างขึ้นมาในชั้นต่าง ๆ และสถาปัตยกรรมลำดับชั้น 7 ชั้น ซึ่งกล่าวถึงฟังก์ชันและการให้บริการในแต่ละชั้น

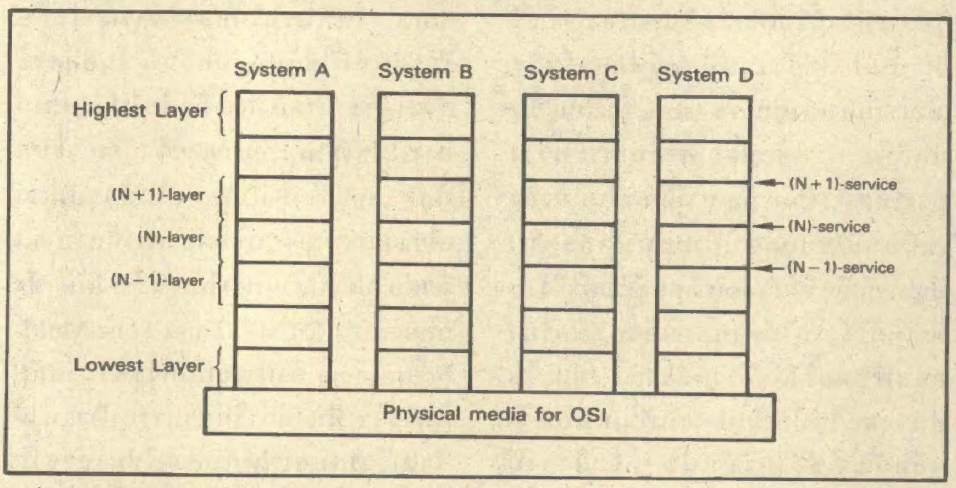
องค์ประกอบของสถาปัตยกรรม

องค์ประกอบของสถาปัตยกรรมเป็นการกล่าวถึงคำจำกัดความและการเขียนสื่อแทนความหมายต่าง ๆ ที่ใช้ในระบบเปิด ดังตัวอย่างที่แสดงในรูปที่ 2 เมื่อกล่าวถึง (N+1)-layer, (N)-layer จะหมายถึงชั้นที่ N+1 และ N ตามลำดับ เมื่อกล่าวถึง (N+1)-service, (N)-service จะหมายถึงการบริการที่อยู่ตรงรอยต่อของชั้นที่ N+1 กับ N+2 และ N กับ N+1 ตามลำดับ

ก. ระบบ (system) หมายถึง ผู้ใช้ เครื่องคอมพิวเตอร์หรืออุปกรณ์ที่สามารถประมวลผลได้ด้วยตัวเอง



รูปที่ 1 สถาปัตยกรรมลำดับชั้น แต่ละชั้นจะมีโปรโตคอลประจำชั้นซึ่งใช้ในการสื่อสารในระดับที่เท่ากันเท่านั้น และแต่ละชั้นจะสื่อสารถึงกันโดยผ่านจุดเชื่อมต่อระหว่างชั้น



รูปที่ 2 ระบบ (system) ชั้น (layer) และบริการ (service)

พร้อมด้วยซอฟต์แวร์และอุปกรณ์รายล้อมที่เกี่ยวข้อง

ข. ชั้น (layer) หมายถึง โครงสร้างเชิงตรรกที่ถูกกำหนดขึ้น เพื่อใช้แบ่งระบบออกเป็นส่วนย่อย ๆ และไม่ขึ้นต่อกัน ทำให้การพิจารณาหรือกระทำกับระบบมีความสะดวกขึ้น

ค. entity หมายถึง ส่วนประกอบภายในชั้นที่ใช้ทำงานต่าง ๆ ภายในชั้นหนึ่ง ๆ จะประกอบด้วยหลาย entity

ง. บริการ (service) หมายถึง ความสามารถที่ชั้นต่ำกว่าให้บริการแก่ชั้นที่สูงกว่าที่อยู่ติดกันดังในรูปที่ 3 เป็นการแสดงให้เห็นการบริการระหว่างชั้นที่ N กับ N+1

จ. จุดให้บริการ (Service Access Point (SAP)) หมายถึงจุดเชื่อมต่อเชิงตรรกระหว่างชั้นที่อยู่ติดกันใช้เป็นจุดสื่อสาร (ให้หรือรับบริการ) ของ entity ระหว่างชั้น รูปที่ 3 เป็นการแสดงให้เห็นว่า (N)-SAP (จุดให้บริการในชั้นที่ N) เป็นจุดเชื่อมต่อระหว่างชั้นที่ N กับ N+1 โดยมีข้อแม้ว่า (N)-SAP หนึ่งจุดจะเชื่อมต่อกับคู่ของ entity ในชั้นที่ N กับ N+1 ได้ในเวลาใดเวลาหนึ่งได้เพียงคู่เดียวเท่านั้น แต่ entity ในชั้นที่ N และ N+1 สามารถเชื่อมต่อกับ (N)-SAP ได้มากกว่าหนึ่งจุด

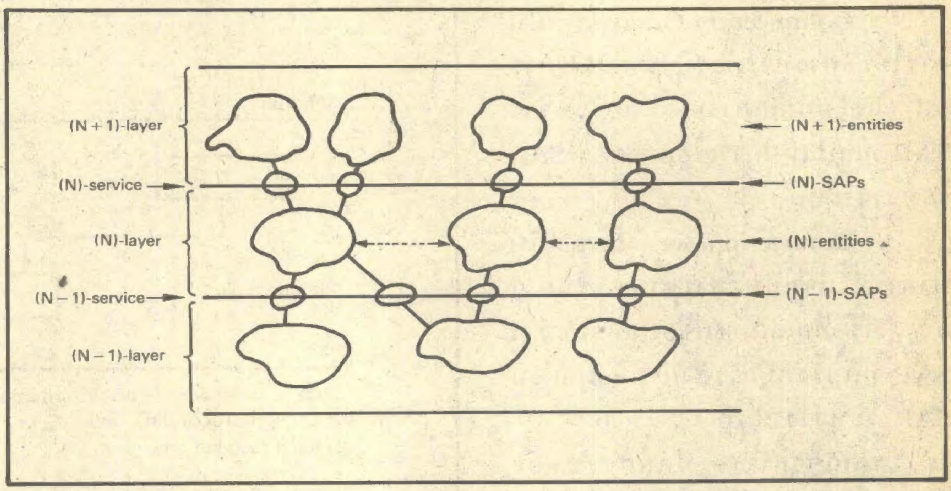
ฉ. ฟังก์ชัน (function) หมายถึง การกระทำที่เกิดจาก entity ภายในชั้นนั้น ๆ

ช. โปรโตคอล (protocol) หมายถึงกฎเกณฑ์และรูปแบบที่ใช้ควบคุมการสื่อสารระหว่าง entity ของชั้นเดียวกันแต่ต่างระบบกัน

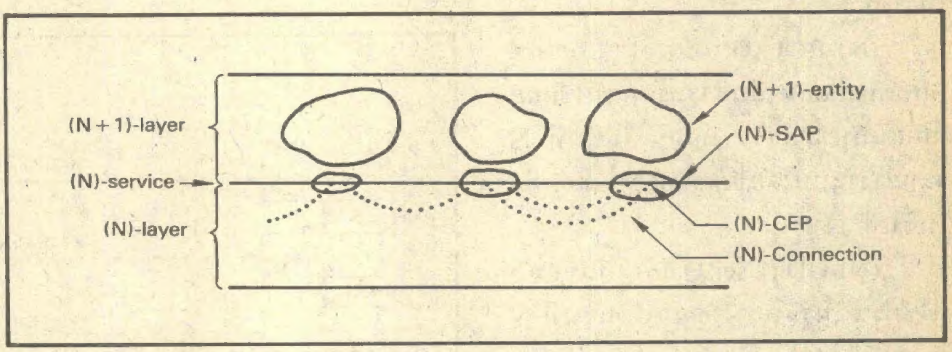
ซ. การกำหนดชื่อ (naming) หมายถึงการกำหนดชื่อของวัตถุใด ๆ ที่อยู่ภายในชั้นหรืออยู่ที่ขอบระหว่างชั้น เช่น entity, SAP เป็นต้น จะต้องมีการกำหนดชื่อให้แน่นอนลงไปเพื่อไม่ให้เกิดความสับสน ตามสถาปัตยกรรมของ OSI ได้กำหนดชื่อของ entity และ SAP ไว้ดังนี้ แต่ละ (N)-entity (entity ในชั้นที่ N) จะถูกกำหนดให้เป็นชื่อสากลเฉพาะ ซึ่งเป็นที่รู้จักกันโดยทั่วไปในระบบเปิด ภายในขอบเขตที่จำกัด N-entity สามารถถูกกำหนดให้เป็นชื่อที่ใช้ภายในขอบเขตนั้น ๆ ได้ แต่ต้องเป็นชื่อที่ไม่ซ้ำกัน และใช้ในขอบเขตนั้น ๆ เท่านั้น แต่ละ (N)-SAP ถูกกำหนดโดย (N)-address ซึ่งจะบอกที่ตั้งเฉพาะของ (N)-SAP นั้น ๆ โดยจะอยู่ระหว่าง ชั้นที่ N และชั้นที่ N+1

ฌ. การเชื่อมต่อ (connection) หมายถึงการเชื่อมต่อระหว่าง (N)-SAP ที่อยู่ในชั้นเดียวกันแต่ต่างระบบกัน การเชื่อมต่ออาจจะเป็นแบบจุดต่อจุดหรือหลายจุดก็ได้ขึ้นอยู่กับบริการนั้น ๆ จุดที่ใช้ในการกำหนดการต่อระหว่าง SAP ทั้งสองหรือมากกว่าคือ (N)-CEP (N-Connection End-Point) แสดงในรูปที่ 4 ซึ่งอยู่ภายใน SAP และถูกกำหนดไม่ซ้ำกัน (N)-CEP ถูกใช้โดย (N+1) และ (N)-entity ของทั้งสองระบบของ (N)-SAP เพื่อกำหนดการเชื่อมต่อในชั้นที่ N

การให้บริการเชื่อมต่อตามสถาปัตยกรรมของ OSI แบ่งออกเป็น 2 ประเภทคือ



รูปที่ 3 entity จุดให้บริการ (SAP) และโปรโตคอล



รูปที่ 4 การเชื่อมต่อ และ CEP

- Connnection-Oriented เป็นการให้บริการที่จะต้องมีขั้นตอนในการขอ และยกเลิกการเชื่อมต่อระหว่าง SAP ทุกครั้งที่มีการเชื่อมต่อของ SAP ระหว่างระบบ

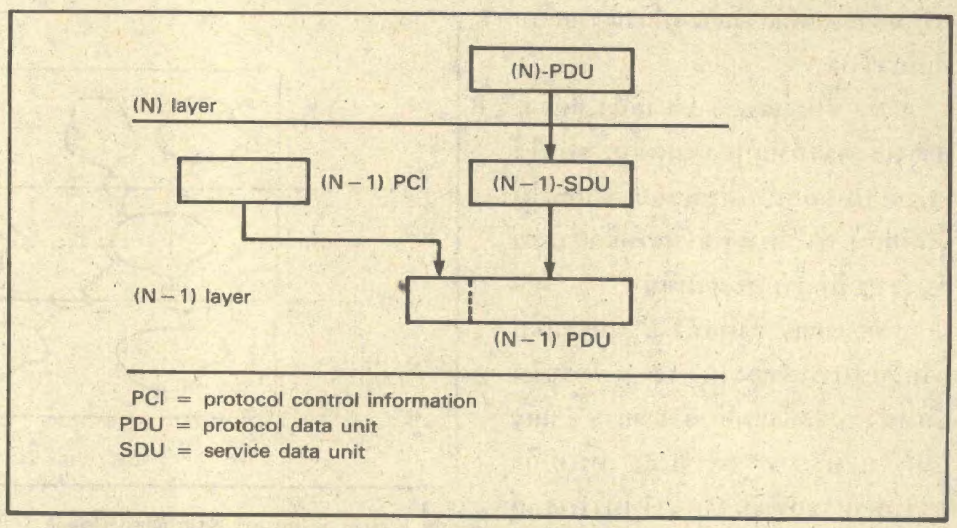
- Connectionless เป็นการให้บริการที่ไม่จำเป็นต้องมีขั้นตอนในการขอ หรือยกเลิกการเชื่อมต่อระหว่าง SAP ทุกครั้งที่มีการส่งผ่านข้อมูลผ่าน SAP ระหว่างระบบ

นอกจากนี้สถาปัตยกรรมของ OSI ยังได้กำหนดรูปแบบของข้อมูลที่ใช้ในการส่งผ่านกันระหว่างชั้นดังแสดงในรูปที่ 5

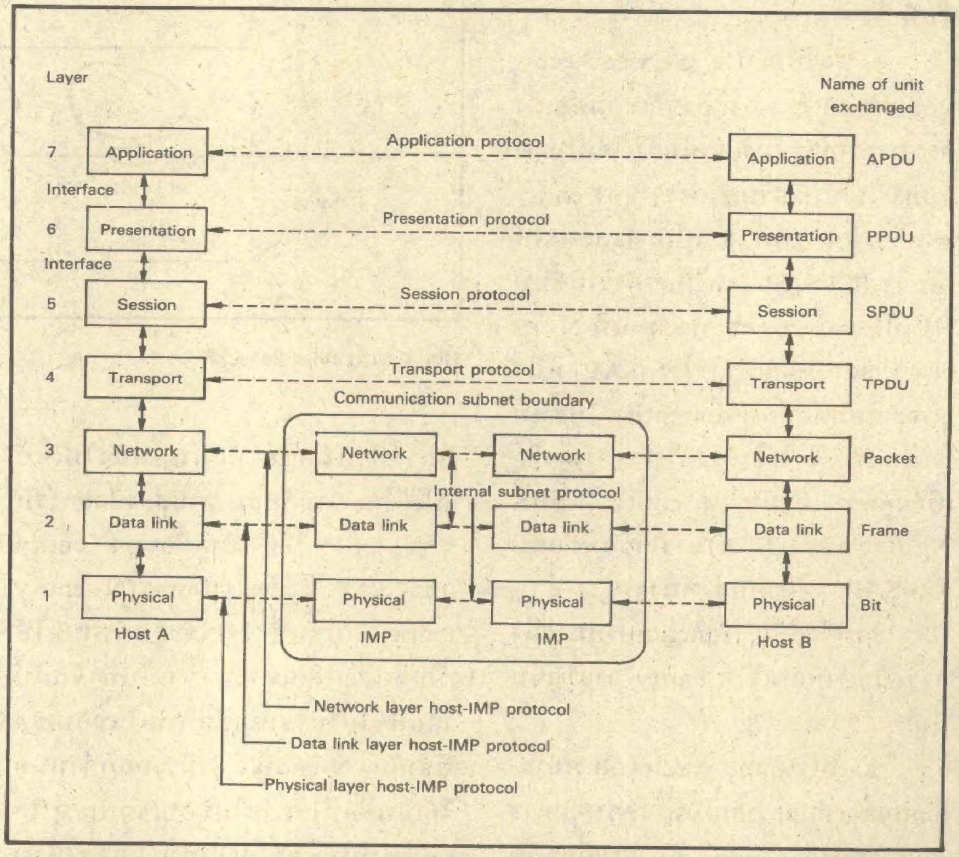
(N)-PCI (Protocol Control Information) เป็นสารสนเทศที่ใช้แลกเปลี่ยนกันระหว่าง entity ในชั้นที่ N ระหว่างระบบโดยใช้การบริการเชื่อมต่อในชั้นที่ N-1

(N)-UD (User Data) เป็นข้อมูลที่ใช้ผ่านกันระหว่าง entity ในชั้นที่ N ในฐานะ entity ในชั้นที่ N เป็นผู้ให้บริการแก่ชั้นที่ N+1

(N)-PDU (Protocol Data Unit) เป็นข้อมูลชุดหนึ่งที่ประกอบไปด้วย (N)-PCI กับ (N)-UD



รูปที่ 5 ความสัมพันธ์เชิงตรรกระหว่างหน่วยของข้อมูลในชั้นที่อยู่ติดกัน



รูปที่ 6 สถาปัตยกรรมลำดับชั้นตามรูปแบบของการเชื่อมต่อระบบเปิด

สถาปัตยกรรมลำดับชั้น 7 ชั้น

สถาปัตยกรรมลำดับชั้น 7 ชั้นคือการแบ่งโปรโตคอลที่ใช้ในการควบคุมการส่งผ่านข้อมูลออกเป็นชั้นย่อย ๆ 7 ชั้น ดังแสดงในรูปที่ 6 โดยแต่ละชั้นจะมีหน้าที่การทำงานแยกจากกันอย่างอิสระ ในกรณีที่ทำงานร่วมกันสามารถติดต่อกันได้โดยผ่านจุดเชื่อมต่อระหว่างชั้นที่อยู่ติดกันเท่านั้น

ก. Physical Layer เป็นชั้นที่ 1 จะกล่าวถึงการเชื่อมต่อทางกายภาพระหว่างเครื่องคอมพิวเตอร์ด้วยกัน เครื่องคอมพิวเตอร์กับเทอร์มินัล หรือเทอร์มินัลกับเทอร์มินัล (DTE/DTE, DTE/DCE) มาตรฐานของชั้นนี้จะกล่าวทางด้าน คุณสมบัติทางกล (mechanical) คุณสมบัติทางไฟฟ้า (electrical) หน้าที่

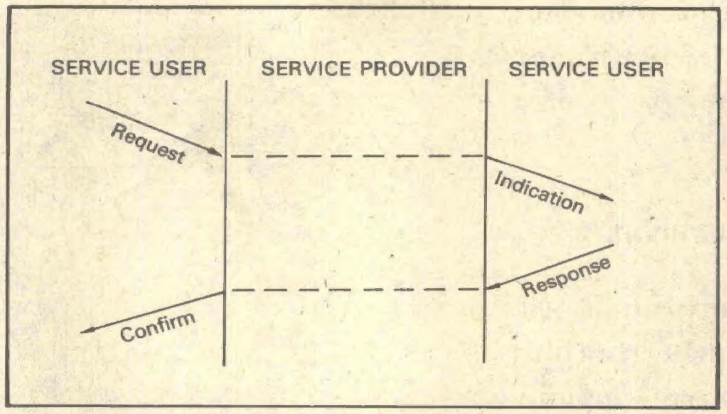
ของการทำงาน (functional) และขั้นตอนในการทำงาน (procedural) ของการเชื่อมต่อ สำหรับข้อมูลในชั้นนี้จะอยู่ในรูปแบบที่เป็นบิต (bit stream)

ข. Data Link Layer เป็นชั้นที่ 2 จะกล่าวถึงการควบคุมและตรวจสอบความผิดพลาดของข้อมูล การควบคุมการส่งผ่านข้อมูล และการให้บริการในลักษณะต่าง ๆ เช่น Connection-Oriented, Connectionless สำหรับข้อมูลชั้นนี้จะถูกจัดให้เป็นกลุ่ม ซึ่งเรียกโดยทั่ว ๆ ไปว่า "เฟรม (Frame)"

ค. Network Layer เป็นชั้นที่ 3 จะกล่าวถึงการควบคุมการทำงานของชุมสายที่ใช้ในการสวิตช์ข้อมูลจากต้นทางไปยังปลายทาง การแก้ปัญหาของการมีข้อมูลภายในระบบมากเกินกว่าที่จะให้บริการได้ สำหรับข้อมูลในชั้นนี้จะถูกจัดให้เป็นกลุ่มซึ่งเรียกโดยทั่ว ๆ ไปว่า "กลุ่มข้อมูล (packet)"

ง. Transport Layer เป็นชั้นที่ 4 จะกล่าวถึงความแน่นอนและความถูกต้องในการส่งผ่านข้อมูลจากต้นทางไปยังปลายทางในลักษณะที่เรียกว่า "end-to-end" คุณภาพของการให้บริการ (quality of service) แก่ session layer ชั้นนี้เป็นชั้นสุดท้ายที่กระทำเกี่ยวกับความแน่นอนในการสื่อสาร (reliable communication)

จ. Session Layer เป็นชั้นที่ 5 จะกล่าวถึงการให้บริการในลักษณะที่เรียกว่า "user-oriented" โดยจะเริ่มมองในแง่ของความสะดวกของผู้ใช้งาน นั่นก็คือ การยอมให้ผู้ใช้ที่อยู่บนเครื่อง (machine) ที่ต่างกันสามารถเปิด session ถึงกันได้ session ที่ว่านี้ได้แก่ remote login การเคลื่อนย้ายไฟล์ เป็นต้น นอกจากนี้ยังให้บริการที่เรียกว่า การซิงโครไนซ์ (synchronization) ซึ่งใช้ในการเรียกสถานะเดิมกลับคืนมาหลังจากระบบเกิดขาดการติดต่ออย่างกะทันหัน



รูปที่ 7
บริการพื้นฐาน 4 ลักษณะ

ฉ. Presentation Layer เป็นชั้นที่ 6 จะกล่าวถึงไวยากรณ์ และความหมาย (syntax and semantic) ของข้อมูล การเข้ารหัสของข้อมูลเพื่อให้ได้อยู่ในรูปแบบมาตรฐานที่สามารถใช้ได้กับทุก ๆ เครื่อง มีการกำหนดโครงสร้างข้อมูลนามธรรม (abstract data structure) นอกจากนี้ยังมีการกล่าวถึง การลดจำนวนข้อมูล (data compression) การเข้ารหัสลับข้อมูล (data encryption) ไว้ในชั้นนี้ด้วย อนึ่ง โครงสร้างข้อมูลนามธรรมที่มีการกำหนดขึ้นมาเป็นมาตรฐาน ISO 8824/CCITT X.208 มีชื่อว่า ASN.1 (Abstract Syntax Notation.1)

ช. Application Layer เป็นชั้นที่ 7 จะกล่าวถึงการให้บริการแก่โปรแกรมประยุกต์ของผู้ใช้งานในระบบ เช่น การกำหนดเทอร์มินัลเสมือน (virtual terminal) เพื่อให้ผู้ใช้บนเครื่องที่ต่างกันสามารถเข้าถึงเทอร์มินัลของกันและกันได้ นอกจากนี้ยังมีมาตรฐานของโปรแกรมประยุกต์ต่าง ๆ เช่น electronic mail, X.400 ของ CCITT หรือ MOTIS (Message Oriented Text Interchange Systems) ของ ISO

ลักษณะของบริการพื้นฐาน

OSI กำหนดลักษณะของการบริการพื้นฐานระหว่างชั้นที่ N กับ N+1 ใด ๆ ดังแสดงในรูปที่ 7 ซึ่งประกอบด้วย 4 บริการพื้นฐาน คือ

ก. request เป็นการที่ผู้ใช้บริการร้องขอการกระทำบางอย่างจากผู้ให้บริการ (ชั้นต่ำกว่าที่อยู่ติดกัน)

ข. Indicate เป็นการที่ผู้ให้บริการร้องขอการกระทำบางอย่างหรือแจ้งการกระทำบางอย่างที่เกิดจากผู้ใช้บริการที่อยู่ในระดับเดียวกันแต่ต่างระบบ

ค. Response เป็นการที่ผู้ใช้บริการตอบกลับสำหรับการร้องขอการกระทำบางอย่างที่จุดให้บริการนั้น ๆ

ง. Confirm เป็นการที่ผู้ให้บริการตอบกลับสำหรับการร้องขอการกระทำบางอย่างที่จุดให้บริการนั้น ๆ

ข้อกำหนดของ OSI สำหรับโปรโตคอลชั้นที่ 2

จุดมุ่งหมายของ OSI ได้กำหนดว่า โปรโตคอลในชั้นที่ 2 จะต้องให้ความเชื่อถือได้ในการส่งผ่านข้อมูลของ

ผู้ใช้ระหว่างระบบ จากข้อกำหนดนี้เองทำให้สามารถอธิบายถึงฟังก์ชันและบริการของโปรโตคอลชั้นที่ 2 อย่างคร่าว ๆ ดังนี้

ฟังก์ชันของโปรโตคอลในชั้นที่ 2

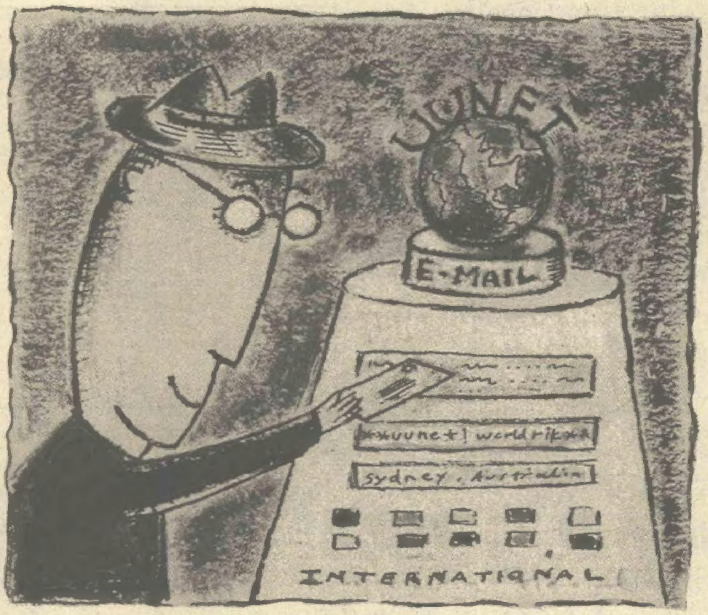
ก. การเริ่มต้นการเชื่อมต่อ (initialization) ใช้สำหรับการสถาปนาการเชื่อมต่อระหว่าง entity ในชั้นที่ 2 ระหว่างระบบไปบนเส้นทางในการส่งผ่านข้อมูลที่ได้ถูกกำหนดขึ้นมาแล้ว ซึ่งเส้นทางนี้อาจจะมีได้มากกว่าหนึ่งเส้นทางก็ได้

ข. การกำหนดจุดปลายทางระหว่างระบบ (identification) ใช้สำหรับกำหนดผู้รับและผู้ส่งข้อมูลระหว่างระบบ เพื่อให้ข้อมูลส่งผ่านกันได้อย่างถูกต้อง

ค. การเข้าจังหวะระหว่างระบบ (synchronization) ใช้สำหรับปรับและสร้างความคงอยู่ให้ผู้รับและผู้ส่งระหว่างระบบ มีกลไกในการเข้ารหัสและถอดรหัสระหว่างกันได้อย่างถูกต้อง

ง. การแบ่งส่วนและการจำกัดขอบเขต (segmenting and delimiting) ใช้สำหรับแบ่งข้อมูลของผู้ใช้ออกเป็นส่วนย่อย ๆ มีขนาดที่แน่นอนและเหมาะสมที่จะใช้ในการส่งผ่านระหว่างระบบ เพราะว่าการส่งผ่านข้อมูลไปบนสื่อนำสัญญาณอาจจะได้รับการรบกวนจากสัญญาณภายนอก ทำให้ข้อมูลไม่สามารถถูกนำไปใช้งานได้ ทำให้ต้องมีการส่งผ่านข้อมูลเดิมจนกว่าจะสามารถนำไปใช้งานได้ ถ้าข้อมูลมีขนาดยาวเกินไปจะทำให้เสียเวลามากในการส่งผ่านข้อมูลซ้ำ แต่ถ้าข้อมูลมีขนาดเล็กจนเกินไปก็ทำให้ไม่มีประสิทธิภาพ

จ. ความโปร่งใสของข้อมูลต่อผู้ใช้ (transparency) ทำให้ข้อมูลในทุก ๆ รูปแบบ (รหัส) ของผู้ใช้งานสามารถส่งผ่านถึงกันได้โดยไม่ต้องคำนึงถึงรหัสพิเศษต่าง ๆ ที่ใช้ในชั้นที่ 2 ว่าจะเป็นอุปสรรคต่อการส่งผ่านข้อมูลระหว่างระบบ



ฉ. การควบคุมการไหลของข้อมูล (flow control) ทำให้ผู้รับสามารถจัดระเบียบการไหลของข้อมูลที่รับจากผู้ส่งเพื่อป้องกันไม่ให้ผู้ส่งส่งข้อมูลมามากเกินไป จนกระทั่งผู้รับไม่สามารถรับหรือตรวจสอบข้อมูลได้ทัน ซึ่งมีผลทำให้เสียเวลาในการเริ่มการรับส่งข้อมูลระหว่างกันใหม่

ช. การควบคุมลำดับของข้อมูลและตรวจสอบความผิดพลาด (error and sequence control) ใช้สำหรับตรวจสอบความผิดพลาดของข้อมูลที่เกิดขึ้นระหว่างการส่งผ่าน ทำให้สามารถแยกแยะข้อมูลที่ผิดออกจากข้อมูลที่ถูกได้อย่างถูกต้อง นอกจากนี้ยังสามารถตอบกลับ หรือแจ้งกลับไปยังผู้ส่งได้ว่ามีเหตุการณ์อะไรเกิดขึ้นเกี่ยวกับข้อมูลนั้น ๆ

ซ. การกลับคืนสู่สถานะเดิม (abnormal condition recovery) ใช้สำหรับตรวจสอบและกลับคืนสู่สถานะเดิมได้อย่างถูกต้อง เช่น ในกรณีที่ระบบไม่ได้รับการตอบรับจากฝ่ายตรงข้าม การส่งผ่านข้อมูลที่มีลำดับไม่ถูกต้อง เป็นต้น วิธีการแก้ไขที่นิยมใช้กันมากคือ การตรวจสอบสถานะภายในเวลาที่กำหนด (time-out)

ฌ. การยกเลิกการเชื่อมต่อ (termination) ใช้สำหรับสิ้นสุดการส่งผ่านข้อมูลระหว่างระบบ การยกเลิกนี้เป็นการยกเลิกการเชื่อมต่อเชิงตรรกเท่านั้น จะไม่รวมถึงการยกเลิกเส้นทางทางกายภาพที่ใช้ในการส่งผ่านข้อมูล

ญ. การบริหารการเชื่อมโยง (link management) ใช้สำหรับดูแลและจัดการการเชื่อมต่อและส่งผ่านข้อมูล เช่น การสถาปนา (establish) การปลดปล่อย (release) และลำดับของข้อมูล (sequence) เป็นต้น นอกจากนี้ยังใช้ในการจัดการกับระบบที่มีรูปแบบของการเชื่อมต่อในลักษณะจุดต่อจุดและหลายจุดด้วย

สถานีในความหมายของโปรโตคอลในชั้นที่ 2 ถูกแบ่งออกเป็น 3 ประเภท คือ สถานีปฐมภูมิ (primary station) เป็นสถานีที่มีสิทธิพิเศษมากที่สุดในโทโพโลยีหนึ่ง ๆ (แบบจุดต่อจุด หรือหลายจุด) เช่น สามารถกำหนดให้สถานีที่เหลือทำงานตามคำสั่ง สามารถส่งหรือรับข้อมูลกับสถานีที่เหลือทั้งหมดได้ ประเภทที่สองคือ สถานีทุติยภูมิ (secondary station) เป็นสถานีที่ถูกควบคุมการทำงานโดยสถานีปฐมภูมิ ในกรณีที่ต้องการส่งผ่านข้อมูลระหว่างสถานีทุติยภูมิด้วยกันเองจะต้องส่งผ่านสถานีปฐมภูมิก่อนเสมอ สถานีประเภทสุดท้ายคือ สถานีผสม (combined station) เป็นสถานีที่รวมความสามารถของสถานีปฐมภูมิและสถานีทุติยภูมิไว้ในสถานีเดียว ดังนั้นในโทโพโลยีที่มีการต่อสถานีผสมเข้าหากันจึงไม่มีสถานีใดมีสิทธิเหนือกว่ากัน

โปรโตคอล HDLC (High-level Data Link Control Protocol)

โปรโตคอล HDLC เป็นโปรโตคอลชั้นที่ 2 ที่ถูกกำหนดขึ้นโดย ISO โดยมีรหัสคือ "ISO 4335" สามารถประยุกต์เข้ากับหลายรูปแบบของการเชื่อมต่อทางกายภาพ และมีหน้าที่การทำงานดังนี้

- ใช้ได้กับการเชื่อมต่อแบบจุดต่อจุดและหลายจุด
- ใช้ได้กับการส่งผ่านข้อมูลระยะใกล้และไกล
- ใช้ได้กับการส่งผ่านข้อมูลแบบฮาล์ฟดูเพล็กซ์และฟูลดูเพล็กซ์
- ใช้ได้กับระบบที่มีการกำหนดประเภทของสถานีทั้ง 3 ประเภท

อนึ่ง ระบบที่มีการกำหนดสถานีปฐมภูมิและสถานีทุติยภูมิ เรียกว่าระบบอสมดุล (unbalanced system) ซึ่งการเชื่อมต่อเป็นได้ทั้งจุดต่อจุดและหลายจุด สำหรับระบบที่ประกอบด้วยสถานีผสมทั้งหมดเรียกว่า ระบบสมดุล (balance system) ซึ่งการเชื่อมต่อเป็นได้เฉพาะจุดต่อจุดเท่านั้น รูปแบบของการส่งผ่านข้อมูลตามข้อกำหนดของโปรโตคอล HDLC แบ่งเป็น 3 ลักษณะ คือ

ก. NRM (Normal Response Mode) ใช้กับระบบอสมดุล สถานีปฐมภูมิต้องการจะติดต่อสถานีทุติยภูมิในกรณีใดก็ได้ แต่สถานีทุติยภูมิจะมีสิทธิ์ติดต่อกับสถานีปฐมภูมิได้ก็ต่อเมื่อสถานีปฐมภูมิอนุญาตเท่านั้น

ข. ARM (Asynchronous Response Mode) ใช้กับระบบอสมดุล สถานีทุติยภูมิสามารถติดต่อกับสถานีปฐมภูมิได้โดยไม่จำเป็นต้องให้สถานีปฐมภูมิอนุญาตก่อน

ค. ABM (Asynchronous Balanced Mode) ใช้กับระบบสมดุล แต่ละสถานีสามารถส่งผ่านข้อมูลถึงกันได้โดยไม่จำเป็นต้องได้รับอนุญาตจากสถานีที่ต้องการจะติดต่อด้วยเสียก่อน

รูปแบบของเฟรม (frame format)

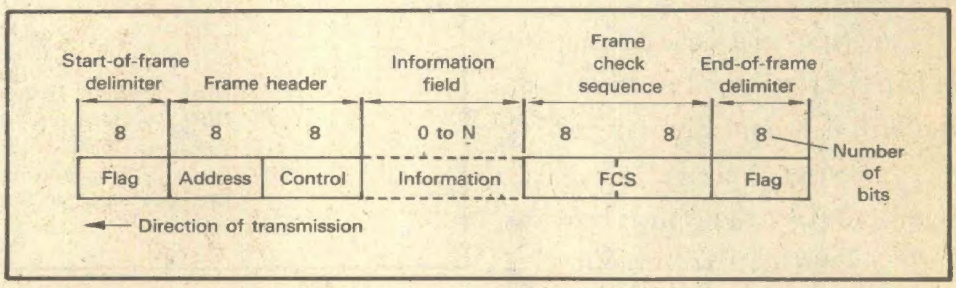
รูปแบบเฟรมของโปรโตคอล HDLC ประกอบด้วย 6 เขตดังแสดงในรูปที่ 8 คือ เขตแฟล็ก (flag field) เขตแอดเดรส (address field) เขตควบคุม (control field) เขตข้อมูล (information field) และเขตตรวจสอบเฟรม (Frame Check Sequence field (FCS)) แต่ละเขตสามารถอธิบายได้ดังนี้

เขตแฟล็ก

เขตแฟล็ก มีขนาด 8 บิต เป็นเขตที่ใช้บอกถึงจุดเริ่มต้นและสิ้นสุดของเฟรม ซึ่งเรียกว่า แฟล็กเปิด (opening flag) และแฟล็กปิด (closing flag) มีรหัสคงที่คือ 01111110_B ($7E_H$)

เขตแอดเดรส

เขตแอดเดรส เป็นเขตที่ใช้บอกถึงหมายเลขของสถานีซึ่งขึ้นอยู่กับว่าเป็นระบบอสมดุลหรือสมดุล ในระบบอสมดุลเมื่อสถานีปฐมภูมิส่งเฟรมไปยังสถานีทุติยภูมิ แอดเดรสนี้จะเป็นหมายเลขของสถานีทุติยภูมิ และถ้าเฟรมนี้ถูกส่งจากสถานีทุติยภูมิ แอดเดรสนี้ก็จะเป็นหมายเลขของสถานีทุติยภูมินั้น ในระบบสมดุล ถ้าเฟรมที่ส่งเป็นเฟรมคำสั่ง (command frame) แอดเดรสนี้จะเป็นหมายเลขของสถานีปลายทาง แต่ถ้าเฟรมที่ส่งเป็นเฟรมตอบสนอง (response frame) แอดเดรสนี้จะเป็นหมายเลขของสถานีที่ส่งเฟรมนั้น เขตแอดเดรสมีขนาด 8 บิตหรือจำนวนเท่าของ 8 บิต โดยมีข้อกำหนดว่าถ้าบิตที่มีนัย



รูปที่ 8 รูปแบบเฟรมของ HDLC

สำคัญน้อย (Least Significant Bit (LSB)) มีค่าเป็น "0" หมายความว่าแอดเดรสถูกขยายออกไปอีก 8 บิต แต่ถ้ามีค่าเป็น "1" หมายถึงเขตของแอดเดรสสิ้นสุดลงที่ 8 บิตนี้ นอกจากนี้ถ้าแอดเดรสมีค่าเป็น 11111111_B (FF_H) จะหมายความว่าเป็นแอดเดรสของทุก ๆ สถานีที่ต่ออยู่ ใช้ประโยชน์ในการกระจายข้อมูลให้กับทุกสถานีโดยใช้แอดเดรสเพียงหมายเลขเดียว

เขตควบคุม

เขตควบคุม เป็นเขตที่ใช้บอกถึงประเภทของเฟรมซึ่งแบ่งออกเป็น 3 ประเภทดังแสดงในรูปที่ 9 คือ เฟรมข้อมูล (information frame) ซึ่งถูกกำหนดโดยบิตที่มีนัยสำคัญน้อยบิตแรกของเขตควบคุมจะต้องมีค่าเป็น "0" เป็นเฟรมที่ใช้ในการรับส่งข้อมูล ประเภทที่สองคือ เฟรมควบคุมและดูแล (supervisory frame) ซึ่งถูกกำหนดโดยบิตที่มีนัยสำคัญน้อยสองบิตแรกจะต้องมีค่าเป็น "10" เป็นเฟรมที่ใช้ในการสอบถามและแสดงสถานภาพการรับข้อมูลของแต่ละสถานี และประเภทสุดท้ายคือ เฟรม unnumbered ซึ่งถูกกำหนดโดยบิตที่มีนัยสำคัญน้อยสองบิตแรกจะต้องมีค่าเป็น "11" เป็นเฟรมที่ใช้ในการส่งผ่านคำสั่ง หรือคำตอบรับในการควบคุมระบบ เช่น การสถาปนาการเชื่อมต่อ การยกเลิกการเชื่อมต่อ เป็นต้น สำหรับส่วนต่าง ๆ ที่อยู่ภายในเขตควบคุมสามารถอธิบายได้ดังนี้

ก. N(S) (send sequence number) ใช้ในการควบคุมการไหลของข้อมูล โดยใช้เป็นค่าที่บอกว่าเป็นเฟรมที่เท่าไร

ข. N(R) (receive sequence number) ใช้ในการควบคุมการไหลของข้อมูล โดยฝ่ายรับใช้เป็นค่าที่บอกว่าคาดว่าจะได้รับเฟรมต่อไปเป็นเฟรมที่เท่าไร หรือเป็นค่าที่บอกว่าเฟรมที่ไม่สามารถรับได้เป็นเฟรมที่เท่าไร สำหรับคำตอบรับหรือคำตอบปฏิเสธตามลำดับ

ค. P/F (Poll/Final) การ poll หมายถึงการที่สถานีใดสถานีหนึ่งส่งคำสั่งไปให้อีกสถานีหนึ่งเพื่อต้องการการตอบกลับ การ final หมายถึงการที่สถานีที่ถูกทำการตอบรับต่อการ poll นั้น ๆ จะเห็นได้ว่าการ poll จะเกิดกับเฟรมที่เป็นคำสั่งเสมอ และจะต้องกำหนดบิต P/F ให้มีค่าเป็น "1" แต่ถ้าเป็น "0" จะไม่เกิดการ poll สำหรับการ final จะเกิดกับเฟรมที่เป็นคำตอบรับเสมอและจะต้องกำหนดบิต P/F ให้มีค่าเป็น "1" แต่ถ้าเป็น "0" จะไม่เกิดการ final

ง. S (Supervisory code) เป็นรหัสที่กำหนดชุดคำสั่งและคำตอบรับดังแสดงในรูปที่ 10 ของเฟรมควบคุมและดูแล

จ. M (Unnumbered code) เป็นรหัสที่กำหนดชุดคำสั่งและคำตอบรับดังแสดงในรูปที่ 10 ของเฟรม unnumbered

อนึ่ง เขตควบคุมจะมีขนาด 8 หรือ 16 บิตก็ได้ขึ้นอยู่กับการกำหนดของระบบนั้น ๆ ในกรณีที่เป็นแบบ 8 บิตจะทำให้สถานีสามารถรับส่งข้อมูลได้สูงสุด 7 เฟรมติดต่อกัน (modulo-8) โดยไม่ต้องรอการตอบกลับจากสถานีตรงข้าม และในกรณีที่เป็นแบบ 16 บิต จะทำให้สถานีสามารถรับส่งข้อมูลได้สูงสุด 127 เฟรมติดต่อกัน (modulo-128) โดยไม่ต้องรอการตอบกลับจากสถานีตรงข้าม

เขตข้อมูล

เขตข้อมูล มีขนาดเป็นจำนวนเท่าของ 8 บิต ใช้สำหรับบรรจุข้อมูล ซึ่งก็คือ PDU ของโปรโตคอลที่อยู่ในชั้นที่สูงกว่านั่นเอง

	CONTROL FIELD BITS							
BIT ORDER	1	2	3	4	5	6	7	8
I frame format	0	N(S)			P/F	N(R)		
S frame format	1	0	S	S	P/F	N(R)		
U frame format	1	1	M	M	P/F	M	M	M

9 (ก)

	EXTENDED CONTROL FIELD BITS															
	1st Octet									2nd Octet						
BIT ORDER	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I frame format	0	N(S)							P	N(R)						
S frame format	1	0	S	S	0	0	0	0	/	N(R)						
U frame format	1	1	M	M	*	M	M	M	F	0	0	0	0	0	0	0

* The value of this bit is undefined

LEGEND

N(S) = Send Sequence Number
N(R) = Receive Sequence Number
P/F = Poll/Final Bit
S = Supervisory Bits
M = Modifier Bits

9 (ข)

รูปที่ 9 การเคลื่อนที่ของหน้าต่าง (ก) เขตควบคุมแบบ 8 บิต (ข) เขตควบคุมแบบ 16 บิต

เขตตรวจสอบเฟรม

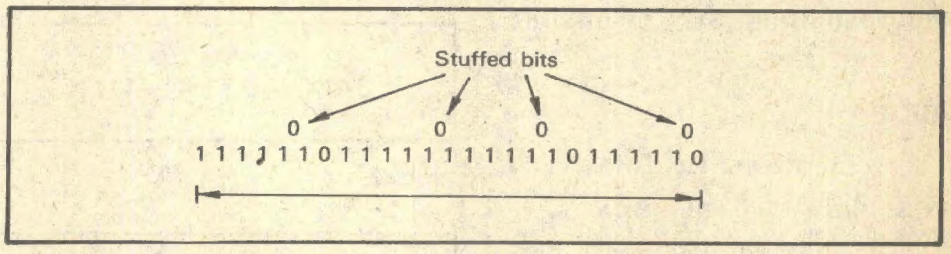
เขตตรวจสอบเฟรม มีขนาด 16 บิต เป็นเขตที่ใช้ในการตรวจสอบความถูกต้องของเฟรมที่ได้รับที่ปลายทางโดยใช้หลักการของ CRC (Cyclic Redundancy Code) สำหรับโปรโตคอล HDLC ใช้ $X^{16}+X^{12}+X^5+1$ เป็น generator polynomial

จากข้างต้นจะเห็นได้ว่า รหัสแฟล็กมีค่าคงที่ ดังนั้นรหัสใด ๆ จะเหมือนกับแฟล็กไม่ได้ เพราะจะทำให้การรับส่งโปรโตคอลผิดพลาดได้ และยังไม่สอดคล้องกับข้อกำหนดที่กล่าวถึง "ความโปร่งใสของข้อมูลต่อผู้ใช้" การแก้ไขปัญหาทำได้โดยวิธีที่เรียกว่าการแทรกบิตศูนย์ (Zero bit insertion) โดยมีข้อกำหนดว่า ถ้ารหัสที่ไม่ใช้แฟล็ก (ยกเว้นรหัส abort และ idle) เมื่อมีบิตที่มีค่าเป็น "1" เกิดขึ้นอย่างน้อย 5 บิตติดต่อกันจะทำการแทรกบิต "0" ลงไปในตำแหน่งที่ 6 เสมอ และบิตที่อยู่ในตำแหน่งถัดไปก็จะถูกเลื่อนไป 1 ตำแหน่งทุก ๆ บิต ดังแสดงในรูปที่ 11 เมื่อปลายทางตรวจพบลักษณะเช่นนี้ก็จะดึงบิต "0" ออกและเลื่อนบิตในตำแหน่งถัดไปเข้ามา 1 บิต ก็จะได้ข้อมูลเดิมกลับมา สำหรับรหัส abort และ idle คือรหัสที่มีบิต "0" และตามด้วย "1" เป็นจำนวน 7 และอย่างน้อย 15 บิตติดต่อกันตามลำดับ

S FRAMES	BITS	
COMMANDS/RESPONSES	3	4
RR — Receive Ready	0	0
REJ — Reject	0	1
RNR — Receive Non Ready	1	0
SREJ — Selective Reject	1	1

U FRAMES		CONTROL FIELD BITS				
COMMANDS	RESPONSES	3	4	6	7	8
SNRM		0	0	0	0	1
SNRME		1	1	0	1	1
SARM	DM	1	1	0	0	0
SARME		1	1	0	1	0
SABM		1	1	1	0	0
SABME		1	1	1	1	0
DISC	RD	0	0	0	1	0
	UA	0	0	1	1	0
SIM	RIM	1	0	0	0	0
TEST	TEST	0	0	1	1	1
XID	XID	1	1	1	0	1
UI	UI	0	0	0	0	0
	FRMR	1	0	0	0	1

รูปที่ 10 รหัสชุดคำสั่งและคำตอบรับ



รูปที่ 11 ลักษณะของการแทรกและดึงบิต "0"

ลักษณะการส่งเฟรมโต้ตอบของโปรโตคอล HDLC

จากรูปที่ 12 เป็นการแสดงให้เห็นการส่งผ่านข้อมูลระหว่างสถานีปฐมภูมิกับสถานีทุติยภูมิโดยมีขั้นตอนดังนี้ สถานีปฐมภูมิต้องการส่งข้อมูลไปยังสถานีทุติยภูมิจึงส่งคำสั่ง SNRM พร้อมกับทำการ poll ไปด้วยเพื่อต้องการให้สถานีทุติยภูมิตอบกลับ เมื่อสถานีทุติยภูมิได้รับเฟรมก็ตอบกลับโดยใช้คำตอบสนอง UA พร้อมกับ final ต่อการ poll ของสถานีปฐมภูมิ หลังจากนั้นสถานีปฐมภูมิก็เริ่มทำการส่งข้อมูล 2 เฟรมติดต่อกัน (เฟรม 0 และ 1) พร้อมกันนั้นก็ทำการ poll ไปในเฟรม 1 เพื่อต้องการให้สถานีทุติยภูมิตอบกลับว่าได้รับเฟรมเรียบร้อยหรือไม่ สถานีทุติยภูมิตอบกลับเป็น RR2 พร้อมกับ final เพื่อแสดงว่ารับข้อมูลได้ถูกต้องทั้งสองเฟรมและคาดว่าจะรับเฟรม 2 ต่อไป เมื่อสถานีปฐมภูมิได้รับคำตอบรับดังกล่าวก็เริ่มส่งเฟรมต่อไป (เฟรม 2 และ 3)

อนึ่ง สถานะในการส่งและตอบรับเฟรมของโปรโตคอล HDLC มีอยู่หลายสถานะ และใช้ได้ทั้งการรับส่งข้อมูลที่เป็นแบบฮาล์ฟดูเพล็กซ์และฟูลดูเพล็กซ์ ซึ่งรายละเอียดหาได้จากเอกสารอ้างอิงท้ายบทความ

FCS การตรวจสอบความถูกต้องข้อมูล

ในการคำนวณหา FCS และการตรวจสอบความถูกต้องของข้อมูล ทั้งภาคส่งและภาครับจะต้องดำเนินการตามขั้นตอนต่อไปนี้

ภาคส่ง

ก. กำหนดค่าในรีจิสเตอร์ของ FCS ให้เป็น "0" หมดทุกบิต

ข. ทำการผกผัน (inverse) 16 บิตแรก (บิตที่อยู่ถัดจากแฟล็กเปิด) ก่อนทำการเลื่อนเข้าสู่รีจิสเตอร์ของ FCS

ค. เลื่อนเขตที่เหลือของเฟรมเข้าสู่รีจิสเตอร์ของ FCS แบบไม่ผกผัน

ง. ทำการผกผันค่าที่อยู่ในรีจิสเตอร์ FCS (เศษของผลหารนั่นเอง) เพื่อบรรจุลงในเขต FCS ของเฟรมส่ง

ภาครับ

ก. กำหนดค่าในรีจิสเตอร์ของ FCS ให้เป็น "0" หมดทุกบิต

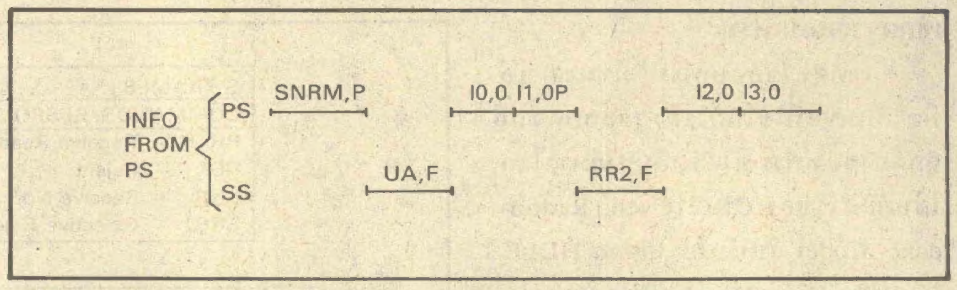
ข. ทำการผกผัน 16 บิตแรก ก่อนทำการเลื่อนเข้าสู่รีจิสเตอร์ตรวจสอบ FCS

ค. ทำการเลื่อนบิตที่เหลือแบบไม่ผกผัน จนกระทั่งถึงบิตเริ่มต้นของเขต FCS

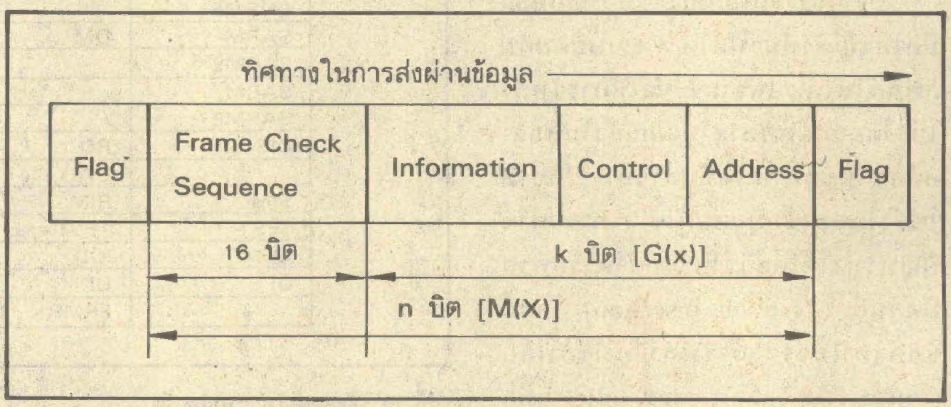
ง. ทำการผกผันค่าของ FCS ก่อนที่จะเลื่อนเข้าสู่รีจิสเตอร์ตรวจสอบ

การใช้ FCS ในการตรวจสอบความถูกต้องของข้อมูลในโปรโตคอล HDLC สามารถแสดงให้เห็นได้ดังรูปที่ 13

การบวกของสมการในหัวข้อนี้เป็นการบวกแบบ modulo-2 คือ 0+0 = 0, 0+1 = 1, 1+0 = 1 และ 1+1 = 0 หมายถึงการบวกแบบไม่มีบิตทด



รูปที่ 12 การรับส่งเฟรมของ HDLC (PS = Primary Station, SS = Secondary Station)



รูปที่ 13 รูปแบบของเฟรม HDLC กับการกำหนดบิต

$$G(X) = X^5+X^3+1\ (1\cdot X^5+0\cdot X^4+1\cdot X^3+0\cdot X^2+0\cdot X^1+1\cdot X^0)$$

รูปที่ 14

$$\frac{X^{16}G(X)+X^k(X^{15}+X^{14}+X^{13}+\ldots+1)}{P(X)} = Q(X)+\frac{R(X)}{P(X)}$$

รูปที่ 15

ในที่นี้กำหนดให้ข้อมูลคือ 101001 สามารถเขียนให้อยู่ในรูปฟังก์ชัน G(X) ได้ดังรูปที่ 14

HDLC ใช้ generator polynomial (P(x)) $X^{16}+X^{12}+X^5+1$ ในการตรวจสอบ ดังนั้น

$$P(X) = X^{16}+X^{12}+X^5+1$$

เมื่อพิจารณาจากขั้นตอนการหา FCS ทางด้านภาคส่งสามารถเขียนสมการได้เป็นดังรูปที่ 15

Q(X) คือผลหารที่ได้ และ R(X) คือเศษที่เหลือจากการหาร ซึ่งต้องนำมาทำให้เป็นค่าผกผันตามขั้นตอนข้างต้นเพื่อทำให้เป็น FCS ดังนั้นสารสนเทศที่ส่งไปยังปลายทาง คือ

$$M(X) = X^{16} G(X) + FCS$$

(ข้อมูลที่ถูกเลื่อนไป 16 บิตกับ FCS)

ทางภาครับมีขั้นตอนคล้ายกับภาคส่ง แต่ต่างตรงที่ข้อมูลที่ถูกพิจารณาทางภาครับคือ M(X) ไม่ใช่ G(X) ดังนั้นในสมการ ในรูปที่ 16 ค่าของ X^k จึงเปลี่ยนเป็น X^n ($n = k + 16$) ซึ่งเขียนเป็นสมการดังรูปที่ 16

Qr(X) และ Rr(x) คือ ผลหารที่ได้ และเศษที่เหลือ ทางภาครับตามลำดับ

ในที่นี้จะแสดงให้เห็นว่าผลที่ได้จากสมการในรูปที่ 16 เป็นค่าคงที่ โดยพิจารณาจากสมการรูปที่ 16 ด้านซ้ายมือ

กำหนดให้ $L(X) = X^{15} + X^{14} + X^{13} + \ldots + 1$ และรูปที่ 17 เป็นการดึงตัวประกอบร่วมออก

การผกผันค่าของ R(X) ในสมการในรูปที่ 15 ให้เป็น FCS คือ การบวกค่าของ R(X) ด้วย L(X) นั่นก็คือ

$$FCS = R(X) + L(X)$$

เมื่อนำค่าของ R(X) และ L(X) แทนใน FCS จะได้ดังรูปที่ 18

จากการบวกแบบ modulo-2 จะได้ดังรูปที่ 19

การตรวจสอบความถูกต้องของข้อมูลนั้นจะไม่สนใจผลที่ได้จากการหาร เพราะเมื่อข้อมูลต่างกัน ผลที่ได้ย่อมต่างกันด้วย แต่จะสนใจเศษที่เหลือจากการหารเท่านั้น ดังนั้นสมการรูปที่ 19 สิ่งที่สนใจคือ

$$\frac{Rr(X)}{P(X)} = \frac{X^{16} L(X)}{P(X)}$$

เมื่อ L(X) และ P(X) ต่างก็เป็นค่าคงที่ ดังนั้นผลที่ได้ก็ย่อมเป็นค่าคงที่ด้วย ซึ่งแสดงให้เห็นว่า ไม่ว่าข้อมูลจะเป็นเช่นไรก็ตาม ถ้าการส่งผ่านข้อมูลถูกต้อง เศษที่เหลือจะเป็นค่าคงที่ตัวเดิมเสมอ ซึ่งก็คือค่าดังรูปที่ 20

อนึ่ง การตรวจสอบความผิดพลาดของข้อมูล โดยใช้ CRC นี้ก็มิได้ให้ผลถูกต้อง 100 เปอร์เซ็นต์กับทุก ๆ กรณีที่มีความผิดพลาดเกิดขึ้นกับข้อมูล ซึ่งรายละเอียดสามารถหาอ่านได้ในเอกสารอ้างอิงท้ายบทความนี้ สำหรับฮาร์ดแวร์ที่ใช้ในการคำนวณ CRC ก็คือ "Feedback Shift Register" ดังแสดงในรูปที่ 21

การควบคุมการไหลของข้อมูล (flow control)

การควบคุมการไหลของข้อมูลเป็นวิธีการที่ใช้ในการควบคุมการส่งผ่านข้อมูล เพื่อป้องกันไม่ให้ฝ่ายส่งส่งข้อมูลมามากเกินไป จนกระทั่งฝ่ายรับไม่สามารถรับข้อมูลได้ทัน ซึ่งมีผลทำให้เกิดข้อผิดพลาดขึ้นได้ โดยทั่วไปฝ่ายรับจะต้องมีการกำหนดบัฟเฟอร์ (buffer) เพื่อรองรับข้อมูลจากฝ่ายส่ง และทำการประมวลผลให้ถูกต้อง หลังจากนั้นจึงจะลบข้อมูลออกจากบัฟเฟอร์

$$\frac{X^{16}(X^{16}G(X) + F(S) + X^n(X^{15} + X^{14} + X^{13} + \ldots + 1)}{P(X)} = Qr(X) + \frac{Rr(X)}{P(X)}$$

รูปที่ 16

$$X^{16}\left[\frac{X^{16}G(X) + FCS}{P(X)} + \frac{X^k(X^{15} + X^{14} + X^{13} + \ldots + 1)}{P(X)}\right]$$

$$X^{16}\left[\frac{X^{16}G(X) + X^k(X^{15} + X^{14} + X^{13} + \ldots + 1)}{P(X)} + \frac{FCS}{P(X)}\right]$$

รูปที่ 17

$$X^{16}\left[\frac{Q(X)}{P(X)} + \frac{R(X)}{P(X)} + \frac{R(X)}{P(X)} + \frac{L(X)}{P(X)}\right] = Qr(X) + \frac{Rr(X)}{P(X)}$$

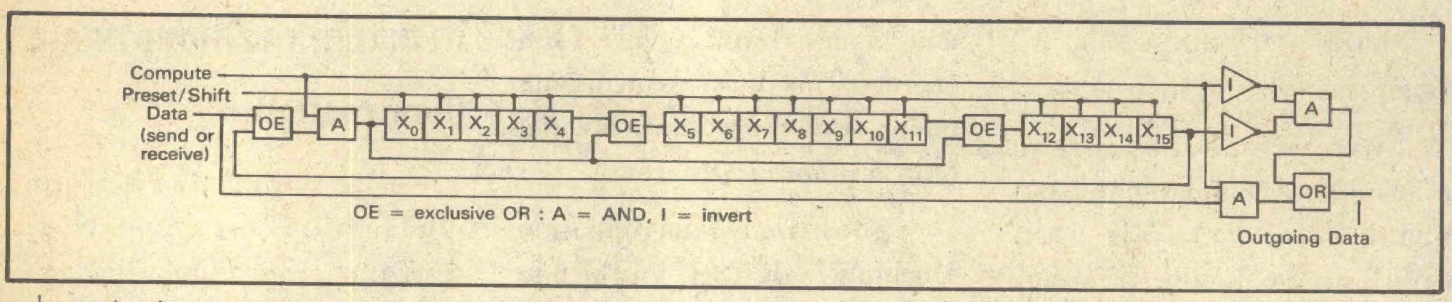
รูปที่ 18

$$X^{16}Q(X) + \frac{X^{16}L(X)}{P(X)} = Qr(X) + \frac{Rr(X)}{P(X)}$$

รูปที่ 19

$$\frac{X^{16}(X^{15} + X^{14} + X^{13} + \ldots + 1)}{X^{16} + X^{12} + X^5 + 1} = 00011101\ 00001111\ (X^{15} \ldots X^0)$$

รูปที่ 20



รูปที่ 21 ฮาร์ดแวร์ในการคำนวณ CRC

เพื่อรอรับข้อมูลใหม่ต่อไป วิธีการที่นิยมใช้เรียกว่า "การเลื่อนหน้าต่าง (sliding window)"

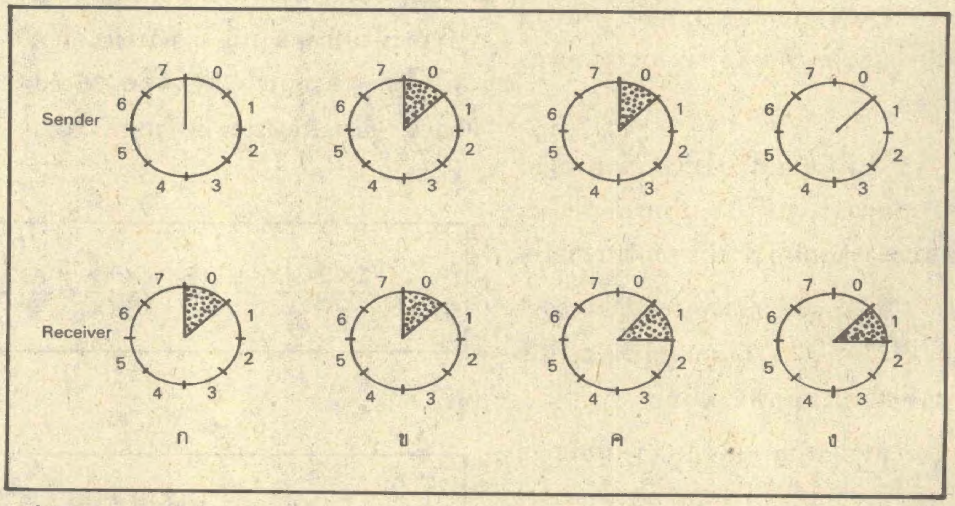
ลักษณะของการเลื่อนหน้าต่างสามารถอธิบายได้ตามรูปที่ 22 โดยแบ่งออกเป็น 2 ส่วนใหญ่ ๆ คือ ผู้รับและผู้ส่ง ทั้งผู้รับและผู้ส่งจะมีส่วนของหน้าต่างทั้งหมด 8 ส่วน (0-7)

- ในขณะที่ผู้ส่งยังไม่ได้ทำการส่งข้อมูลหน้าต่างของผู้ส่งยังปิดอยู่ ในที่นี้กำหนดให้ผู้รับสามารถรับข้อมูลได้ครั้งละ 1 ส่วน จึงทำให้ผู้รับเปิดหน้าต่างอยู่ 1 ส่วนพร้อมที่จะรับข้อมูลจากผู้ส่งดังแสดงในรูปที่ 22 ก.

- เมื่อผู้ส่งทำการส่งข้อมูลออกไปก็จะทำการเปิดหน้าต่างตัวเองออกมา 1 ส่วน ขณะเดียวกันผู้รับก็รอรับข้อมูลจากผู้ส่งดังแสดงในรูปที่ 22 ข.

- เมื่อผู้รับรับข้อมูลจากผู้ส่งได้ถูกต้องก็จะทำการเลื่อนหน้าต่างของตัวเองออกไป 1 ส่วน (จาก 0 เป็น 1) พร้อมที่จะรับข้อมูลส่วนต่อไป ขณะเดียวกันผู้ส่งก็รอรับการตอบกลับจากผู้รับดังในรูปที่ 22 ค.

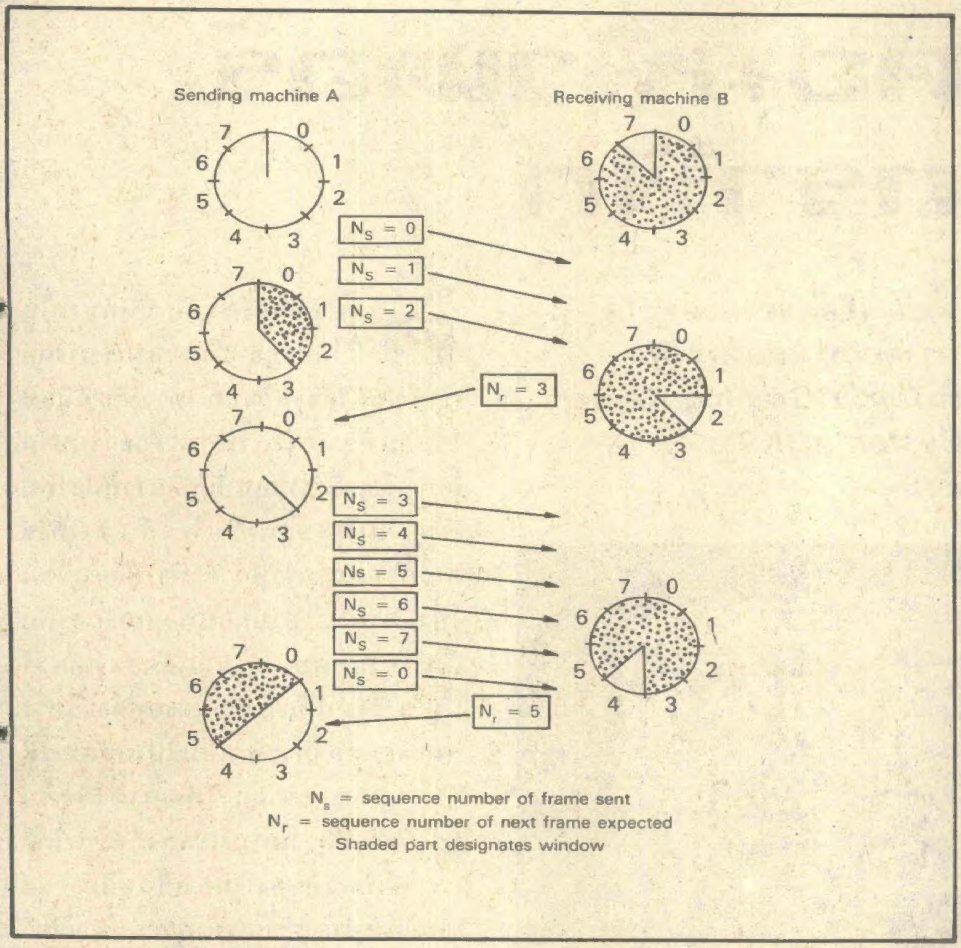
- เมื่อผู้ส่งรับทราบว่าข้อมูลส่งได้ถูกต้องแล้วก็ทำการปิดหน้าต่างเดิมลงและพร้อมที่จะส่งข้อมูลต่อไป (ถ้ามีข้อมูลส่ง) ขณะเดียวกันผู้รับก็พร้อมที่จะรับข้อมูลส่วนต่อไปดังแสดงในรูปที่ 22 ง.



รูปที่ 22 การเคลื่อนที่ของหน้าต่าง

จากขั้นตอนข้างต้นจะเห็นได้ว่าผู้ส่งจะเปิดหน้าต่างได้ก็ต่อเมื่อมีข้อมูลจะส่ง และจะปิดหน้าต่างได้ก็ต่อเมื่อผู้รับได้แจ้งกลับมาว่าได้รับข้อมูลเรียบร้อยแล้ว ส่วนผู้รับจะเปิดหน้าต่างคอยข้อมูลจากผู้ส่งเสมอและจะเปลี่ยนช่องหน้าต่างก็ต่อเมื่อได้ตรวจสอบข้อมูลถูกต้องพร้อมกับแจ้งกลับไปยังผู้ส่งแล้วเท่านั้น จากรูปที่ 22 ถ้าคิดต่อไปจะพบว่าผู้รับไม่สามารถเปิดหน้าต่างได้มากกว่า จำนวนหน้าต่างสูงสุด-1 ที่เป็นเช่นนี้ก็เพราะว่าถ้ามากกว่าหรือเท่ากับ แล้วจะทำให้เกิดการทับซ้อนหน้าต่างเดิม ทำให้ผู้ส่งและผู้รับไม่สามารถตรวจสอบลำดับของข้อมูลได้อย่างถูกต้อง เพื่อให้มีความเข้าใจได้มากขึ้นจะขออธิบายตัวอย่างเพิ่มเติมในรูปที่ 23

จากรูปกำหนดให้ผู้ส่งและผู้รับสามารถส่งรับข้อมูลได้มากที่สุด 7 เฟรมติดต่อกัน ในช่วงแรกผู้ส่งทำการส่ง 3 เฟรม (N=0, 1, 2) ทำให้หน้าต่างของผู้ส่งถูกเปิดออก 3 ส่วน เมื่อผู้รับรับเฟรมได้ถูกต้องจึงเลื่อนหน้าต่างของตัวเองให้ไปอยู่ที่ตำแหน่งใหม่พร้อมกับส่งข่าวสารกลับไปยังฝ่ายส่งให้เริ่มส่งเฟรมที่ 3 มาได้ เมื่อฝ่ายส่งได้รับการยืนยันจึงปิดหน้าต่างลง 3 ส่วน ทำให้ตำแหน่งเริ่มต้นของหน้าต่างจะเปลี่ยนไปจากเดิม หลังจากนั้นก็ทำการส่งข้อมูลเพิ่มเติมไปอีก 6 เฟรม (3 ถึง 7 และ 0) แต่ในที่นี้ฝ่ายรับต้องการที่จะจำกัด



รูปที่ 23 การควบคุมการไหลของข้อมูล

การส่งข้อมูลของฝ่ายส่งซึ่งอาจจะเนื่องมาจากฝ่ายรับมีบัฟเฟอร์ไม่เพียงพอ จึงตอบกลับไปยังฝ่ายส่งให้เริ่มส่งตั้งแต่เฟรมที่ 5 กลับมาใหม่

จากข้างต้นจะเห็นได้ว่าเป็นการยกตัวอย่างของการเลื่อนหน้าต่างสำหรับการส่งผ่านข้อมูลแบบฮาล์ฟดูเพล็กซ์ เพื่อให้ง่ายต่อการเข้าใจ ซึ่งตามที่เป็นจริงแล้วการเลื่อนหน้าต่างสามารถประยุกต์ได้กับการส่งผ่านข้อมูลแบบฟูลดูเพล็กซ์เช่นกัน

บทสรุป

มาตรฐานของระบบเครือข่ายสื่อสารข้อมูลในปัจจุบันมีเป็นจำนวนมาก ดังนั้นในการศึกษาจึงควรจะมีมาตรฐานที่เป็นรูปแบบทั่วไปและเป็นที่ยอมรับ ไว้เป็นพื้นฐานของแนวความคิด เพื่อจะได้ศึกษามาตรฐานอื่น ๆ ต่อไปได้โดยไม่เกิดความสับสน การศึกษาโปรโตคอลก็เช่นเดียวกัน ในที่นี้ยกตัวอย่างของ HDLC ก็เพราะว่า มีการนำ HDLC ไปเป็นแบบอย่างของโปรโตคอลในชั้นที่ 2 ของมาตรฐานอื่น ๆ อยู่พอสมควร เช่น LAPB (Link Access Procedure Balance) ใน X.25, LAPD (Link Access Procedure in D-Channel) ใน ISDN และ LLC (Logical Link Control) ในอนุกรม IEEE 802.X เป็นต้น

เอกสารอ้างอิง

1. Tanenbaum A.S., Computer Network, PHI, New Jersey, 2 nd ed., 1989

2. เขมะฑัต วิภาตะวนิช, "การออกแบบและสร้างเครื่องมือตรวจสอบโปรโตคอลชนิด SDLC," วิทยานิพนธ์ปริญญามหาบัณฑิต, ภาควิชาวิศวกรรมไฟฟ้าบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2530

3. Folts, H.C., Data Communication Standards, McGraw-Hill, New York, vol I, II, III, 1986.